Machine Learning Engineer Nanodegree Capstone Proposal - Dog Breed Classifier Using CNN

Manish Chandra
September 12, 2020

1 Domain Background

Recognising dogs according to their respective breed is a challenging task even for humans. There are hundreds of breeds in existence which are grouped into 10 distinct groups according to physical characteristics. This shows that such classification is a complex task, as remembering all the breeds and distinguishing between similar breeds cannot be easily done by humans. This is a multi-class classification problem where we can use a supervised learning approach. Image classification refers to a task in computer vision of classifying an image according to its visual content. A convolutional neural network (CNN) has shown great success in the image domain. CNNs are a specialized type of neural network for processing data that have the structure of a grid.

2 Problem Statement

The task is to build a pipeline to process real world, user-supplied images. Given an image of a dog, the algorithm should give an estimate of the dog's breed. And when the image is of a human, the algorithm should give an estimate of a dog's breed that resembles the human in the image. If neither a dog nor a human is detected in the image, the algorithm should output an error message. Furthermore, the model should achieve 60 percent or greater accuracy.

3 Datasets and Inputs

The dataset for this project is provided by Udacity.

- Dog images dataset: The dog image dataset has 8351 total images which are split into train (6,680), valid (835 Images) and test (836). There are a total of 133 dog breeds and each breed has a separate folder for it. The images are of different sizes and backgrounds.

- Human images dataset: The human image dataset contains 13233 total human images inside a total of 5750 folders. All the images are of size 250x250. Images have different backgrounds.

4 Solution Statement

For performing this multi-class classification, a CNN model using transfer learning will be trained that can estimate as accurately as possible the breed of a dog that is there in the input picture. To this end, whether a human or a dog is there in the picture needs to be identified beforehand. If a human is present, the resembling dog breed will be predicted. If a dog is present, an estimate of the dog breed will be predicted.

5 Benchmark Model

A CNN model built from scratch having accuracy of at least 10% (much better than random guessing) will be used as the benchmark model.

6 Evaluation Metrics

Accuracy will be used for evaluation and comparing different models.

7 Project Design

Step 0:
The datasets for the human and dog images will be imported

Step 1:
OpenCV's implementation of Haar feature-based cascade classifiers will be used to detect human faces from the datasets.

Step 2:
The pre-trained VGG-16 model (trained on the ImageNet dataset) will be used here to design a dog detector.

Step 3:
Image augmentation will be performed on the training data. A CNN model will be created from scratch to predict dog breed and its performance on test data will be recorded.

Step 4:
CNN model will be created using transfer learning to predict dog breed. The augmented training data will be used in this step as well. Based upon the performance of different architectures on ImageNet[2], ResNet-152 will be used as a feature extractor (base network).

Step 5:
Everything will be put together in this step. With a new image as input, whether it has a human or a dog will first be identified. If a dog is detected, the predicted breed will be returned. If a human is detected, the resembling dog breed will be returned. If neither is detected, then an error message will be displayed.

Step 6:
The algorithm from step 5 will be tested on the images of my choosing to assess the final model.

References:

1. *Original repo with starting code: https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification*
2. *https://pytorch.org/docs/stable/torchvision/models.html*